AD-A084 127 RENSSELAER POLYTECHNIC INST TROY N Y F/G 5/8
THE RADC INTERACTIVE LABORATORY FOR DESIGN OF PATTERN RECOGNITI--ETC(U)
1975 J C FAUST, H E WEBB, L A GERHARDT AFOSR-73-2486

UNCLASSIFIED NL

1 OF 1
AD
A084127

END
DATE
FILMED
6-80
DTIC

(12) 30L

(1)

THE RADC INTERACTIVE LABORATORY FOR DESIGN OF PATTERN RECOGNITION SYSTEMS AND ITS APPLICATION*

(11) 1975

John C. Faust
Haywood E. Webb, Jr.
Rome Air Development Center
Griffiss AFB, New York

And

Lester A. Gerhardt
Rensselaer Polytechnic Institute
Troy, New York

(10) John C./Faust Haywood E./Webb, Jr. Lester A./Gerhardt [redacted]

This paper describes the Interactive Laboratory for Design of Pattern Recognition Systems which exists at the Rome Air Development Center (RADC) of the United States Air Force. A brief history of the research that led to the interactive approach is included, together with the philosophy of the interactive approach. Applications of the laboratory to some real problems are discussed, together with some comments on its use in a course in Pattern Recognition given at RADC. The paper is tutorial in the sense that most of the results have been previously published in fragments. The main contribution of this paper is a description of a real physical laboratory whose implementation is based on an interactive approach to pattern recognition which has evolved over the years.

(15) AFOSR-73-2486

1. Introduction

A classifier is a function C whose domain is the input measurement space and whose range is the set of classes or categories. If class conditional densities are defined over the measurement space together with the usual assumptions of classical decision theory, the function C can be found by invocation of Bayes Theorem. For this case, the function C, and the physical device which realizes C are optimum in the sense of minimum Bayes Risk.

In many important real world classification problems, the class conditional densities over the measurement space are not known. In this paper it is assumed that "representative" data samples in measurement space are available, however, and that the samples are labeled by class or category. This is fundamentally a nonparametric approach. In this approach, there is the necessity that the classifier designer study the problem to learn about the data through experiments conducted on a large number of these representative samples, together with available a priori knowledge of the "phenomenology" of the problem. To improve the efficiency of this human learning process, an

* Paper presented at the 1975 Conference on Computer Graphics, Pattern Recognition and Data Structure at Beverly Hills, California on 15 May 1975.

interactive approach has been chosen. The basic philosophy is to couple the man and the machine as a team so that each can contribute what it can do best. The man can contribute his intelligence, and his knowledge, about the problem. The machine can contribute its ability to do bookkeeping, complicated calculations, and display results on a graphics terminal in forms readily interpretable by the man.

Though this paper focuses upon the particular interactive laboratory for the design of pattern recognition systems implemented at RADC, some other interactive systems for similar purposes are enumerated in Table 1.

SYSTEM NAME	DEVELOPER
SARF	General Motors Corp.
DX-1	AF Cambridge Res Lab
INTERSPACE	Purdue University
IFES	USAF (RADC)
Merlin System	Merlin Systems Corp.
IBM Interactive Sys	IBM Corp.

TABLE 1 - Other Interactive Pattern Recognition Systems

More details on these systems may be found in Kanal[1].

This paper consists of eight sections. The remainder of this section consists of a brief history of the pattern recognition research conducted at RADC during the past sixteen years, and the scope of the present laboratory. Section 2 discusses the philosophy of the interactive approach to the design of pattern recognition systems. Section 3 presents a functional overview of the Waveform Processing System (WPS) which is used for waveform data analysis and feature extraction. Section 4 gives a description of the On-Line Pattern Analysis and Recognition System (OLPARS) and contrasts the two different implementations of OLPARS at RADC. Section 5 documents additional elements of the Laboratory, and section 6 discusses various applications of the Laboratory. Some elements of the Laboratory have been used for laboratory experiments in a short course in Pattern Recognition. Section 7 comments on this experience. Finally, some comments on the number of data samples needed to design reliable classification logic are presented in section 8.

To obtain some idea of the scope of the Laboratory and how the interactive

approach was selected, the history of its development will be briefly reviewed. Contributions to this development were made by many individuals and organizations sponsored by RADC. The list of contributors and their specific contributions is too long to be mentioned here, but these contributions are acknowledged to be an integral part of the ideas which led to what is now the Laboratory.

Work in pattern recognition research at RADC began in 1959 with joint sponsorship of the PERCEPTRON with the Office of Naval Research. It ultimately became clear that the single layer PERCEPTRON could adaptively construct only linear boundaries. From the knowledge that linear separable problems formed only a small subset of the real problems, work was sponsored on the multi-layered PERCEPTRON due to its ability to construct piecewise linear boundaries. This research was directed to finding algorithms for the adaptive construction of an optimum piecewise linear boundary. This problem turned out to be untractable. Subsequently, the search for other structures and convergence algorithms was made using automata theory, computability theory, and a theory of self-organizing systems on the one hand, and parametric statistical ideas on the other. All of these concepts were considering the general idea of a universal adaptive or learning device which, when given a sufficiently large number of labeled data samples, would converge to the optimum classifier.

In 1966 the Mattson-Dammann algorithm[2] for pattern classification was implemented on the CDC 1604 computer for use in an interactive mode with the Bunker Ramo BR-85 display console. This preliminary interactive pattern recognition system was called the DOCUS (Display Oriented Computer Usage System) Pattern Recognition Overlay.[3]

By 1968, based on experience with DOCUS together with results from other research programs, three conclusions were apparent:

(1) The classification design procedure should be interactive with emphasis on the learning in the problem being done by man instead of the machine.

(2) The system should contain a menu of algorithms instead of relying on a single algorithm.

(3) Structure analysis of data should precede classifier design.

Further experiments through 1970 tended to confirm the above hypotheses.

A system, OLPARS, was defined by Sammon[4] in 1968 for the solution of pattern analysis and pattern classification problems using an interactive, graphics oriented computer system. Implementation of OLPARS began on the CDC 1604 computer and the BR-85 display console in 1968 and was completed in 1971.[5] Subsequently, this system was used in the solution of several pattern

recognition problems. Some of these problems are described by Simmons[6] others are listed in section 6.

Also in 1968 the need for interactive feature definition and extraction systems was recognized.[4] The elements of the current Laboratory were defined in 1970. In addition to OLPARS, it contains an interactive feature definition and extraction system for waveform data.[7,8] It is these items upon which we will focus in this paper. The realization of this laboratory represents an investment on the order of 25 man years and over $700,000 in hardware.

2. The Interactive Approach to Pattern Recognition

Since the advent of the general purpose digital computer, there has been a growing interest in producing machines which are capable of duplicating the recognition and decision making functions previously reserved for humans. The relevant body of knowledge which has been generated as a result of this interest has been called pattern recognition theory. We may define pattern recognition as the automatic classification of the state of an environment based upon a set of measurements made on that environment. Hence, solutions to the general pattern recognition problem involve solutions to the problems of data collection and pattern classification, as depicted in Figure 1.

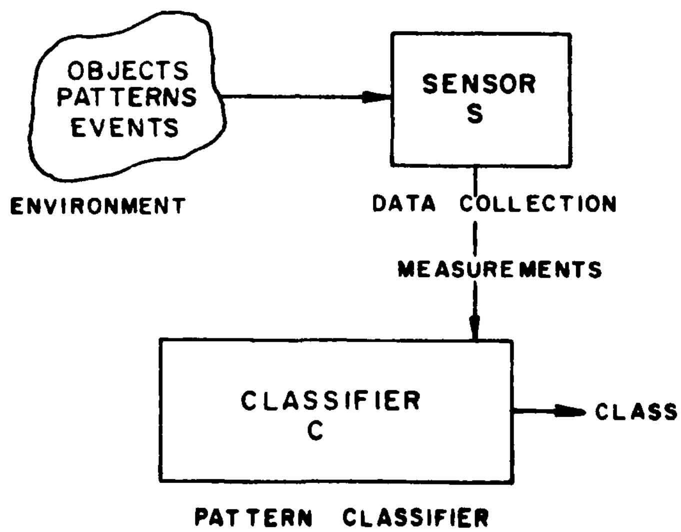

FIGURE 1 - General Pattern Recognition Problem

It is the usual procedure to design the classifier C by a cascade of two functions. The first of these functions is called a feature extractor. This feature extractor is a function F whose domain is measurement space, and whose range is a space called feature space. The second function C' is a mapping whose domain is feature space and whose range is the set of classes. Figure 2 illustrates this concept.

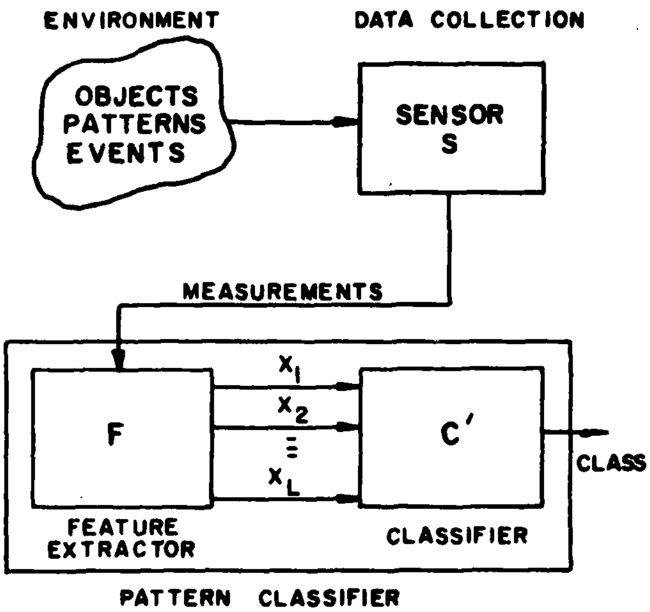

FIGURE 2 - General Pattern Recognition Problem
Illustrating Internal Structure of Pattern Classifier

It is observed that C and C' are both classifiers having different domains, but the same range.

$$C = C'(F) \qquad (1)$$

However, the representation of C in (1) is not unique, so that many realizations of this equation are possible.

The approach of Figure 2 is taken based on the following observations. For most pattern recognition problems:

a. At best only partial "a priori" information is available.

b. Data samples labeled by class are available.

c. When the measurement space represents images or waveforms, the dimensionality (the number of digital measurements) is large; e.g., > 100.

In the absence of sufficient "a priori" information to specify the form of the optimum classifier, or even one whose performance approximates that of the best, we must take an empirical approach to the solution of pattern recognition problems. Hence, given a sufficient number of labeled data samples (see Section 8) one approach would be to design many different classifiers on an empirical basis, compare them, and choose the best. However, the number of potential classifiers under this approach is so large, that to define each, and compare them to select the best would not be computable. Somehow the additional information provided by the labeled data samples must be used in an efficient manner, so that the number of potential candidate classifiers is not too large, and yet hopefully includes the best classifier or at least one that reasonably approximates it. Any method to generate "reasonable" candidates must be based on whatever "a priori" information is available coupled with any additional insight which can be gained by the designer during problem solution. Since this insight must be obtained from the labeled data samples, the designer must have the ability to observe properties of the data in measurement space. Interaction between the designer and the data, using the scientific method to gain this insight, shows high promise. In this case, it is the designer, rather than an adaptive classifier, who learns and obtains insight about the problem. The man embodies what he has learned into the classifier design. This is what we call the interactive approach. To successfully use it, one must iterate several aspects or pieces of the problem several times.

The concept of a vector space is fundamental in the solution of pattern recognition problems. The measurements made by the sensor on a given object in the environment can be represented as a vector in measurement space. If the sensor output is a string of digital numbers, this is clearly the case. When the measurements are either waveforms or images, it is a classical result that this is so. Similarly, the features obtained from the feature extractor define the basis of a vector space, and an object or an event is represented as a vector or point in this space. If we have extracted L features, then each object is represented as a point in L-dimensional feature space. Thus, feature extraction can be viewed as a transformation (in general, non-linear) from the measurement vector space to the feature vector space. Pattern classification defines the partitionment of a vector space (the measurement space or feature space) into regions associated with each of the states (classes) of the

environment. In order to solve a pattern recognition problem, sample vectors for each state (class) must be collected and analyzed in order that a satisfactory pattern classifier be designed.

In many cases, however, the data collected is in the form of waveforms, two-dimensional imagery or a large number of digital measurements. The function of feature extraction then, is to map each object described by the raw data into a useful smaller set of discriminating features. They are normally selected under the criterion that they possess only the essential information necessary for discrimination between classes, rather than a complete description of the characteristics of the given classes.

Once a candidate set of features has been extracted, we proceed to the pattern classification problem. Before proceeding to define the boundaries of the classification regions (i.e., designing the recognition logic), however, we first ask the question: Do the features selected adequately distinguish between the classes to be recognized? Hence, we first determine whether the data points for each class tend to cluster or group together in the vector space defined by the features (pattern analysis). If they do, then we can proceed to design the classification logic; if they do not, then we must return to the feature extraction stage, and extract a better set of features before continuing.

In the preceding discussion, we have seen that the rationale for an interactive approach resulted from the lack of sufficient "a priori" information necessary to specify the form of the classifier in a straightforward manner for most real-world pattern recognition problems. Based on this fact, the desirability of an interactive, graphics oriented approach to the design of pattern recognition systems can be further substantiated as follows:

a. Feature extraction procedures are dependent upon the form and type of raw data, and the particular recognition problem at hand; on the other hand, no single algorithm or procedure exists which is capable of solving all pattern classification problems. Therefore, an organized collection of different techniques in the form of a menu seems appropriate. This organization should permit the addition of new techniques to the menu.

b. A wide variety of efficient and flexible techniques for data handling, visual inspection and numerical computation should be available to the operator/design engineer.

i) An efficient filing system for handling large amounts of sample data is necessary so that a sufficient sample size for both the design and test data sets can be achieved, thus improving the reliability of the resulting classification logic.

ii) Suitable graphics is necessary to exploit the human's ability to

recognize data structure in high-dimensional vector data (e.g., clusters), and candidate features in waveform or image data.

iii) Not only should the choice of any technique within the system be under operator control, but also the choice of parameters for executing a particular technique once it has been chosen.

c. To aid and stimulate the human designer in invoking the scientific method, the time delay between the initiation of a request and its completion should be compatible with the operators thought processes, or least be short enough that it will not interrupt his train of thought.

d. Finally, for completeness, we mention the point we stressed earlier. The boundaries between feature extraction and pattern classification are not sharp. An empirical solution to a pattern recognition problem invariably involves repeated iteration between both in a manner which cannot be predetermined.

Hence, the pattern recognition problem solver must be provided with an easy-to-use, flexible interactive computer system, which provides him with an efficient means for applying and evaluating a wide variety of algorithmic techniques for feature extraction and decision logic design to large quantities of data.

3. The Waveform Processing System (WPS)

WPS is an interactive, graphics oriented computer system for the extraction of features from waveform data and the analysis of a waveform data base. Its chief purpose is to provide the analyst with a library of mathematical algorithms and display options he can call upon from the display console, so that he can design and evaluate feature extraction techniques for waveform pattern recognition problems. Once a set of features have been extracted from each of the members of a waveform data base, the analyst can input them into the OLPARS system to begin the pattern classification logic design phase of the problem solution.

One idea which we believe will significantly contribute to the feature extraction problem is the direct invocation of the scientific method of observation, hypothesis formulation, and experimental verification of hypothesis.

WPS is the physical realization of a system to make this idea practical. WPS permits the man to observe waveform pictures of the data. The man forms hypotheses about features he proposes. WPS provides the man with a tool for rapidly testing these hypotheses. It is by the iteration of this process that suitable features will be found if they exist. A priori information may still be used; although trial and error procedures are not completely eliminated, it

is believed that they will be considerably reduced by the human insight gained during the iterative process.

The Waveform Processing System (WPS) is currently being implemented on a DEC PDP-11/45 computer with a Vector General display and control console, and a Tektronix 4002A storage tube with a hardcopy unit for hardcopying selected Vector General displays. Implementation is expected to be completed in September 1975. The description given here is as it currently is conceived and, therefore, is not complete in details.

WPS has been designed in a modular fashion to provide a large degree of flexibility. It is comprised of four software modules: the WPS Executive, the WPS Filing System, the Waveform Display Modules, and the Applications Programs.

The firot three modules are in core during normal operation of the system. The fourth module operates as a software overlay with specific applications programs being swapped into core upon request.

The WPS Executive

The WPS Executive provides the basic interface for all the system modules and coordinates all system activities. The analyst, seated at the user console, makes his requests known to the system by keying in commands through the user console keyboard. After the executive receives a request, it interprets the request and then loads the necessary applications program or data from the appropriate modules.

The options available to the WPS user consist of a sequence of frames linked together in the form of a hierarchical control tree. Up to sixteen options are available on each frame. Figure 3 indicates how these frames are structured in the tree. Selection of any option on a given frame is accomplished by depressing the corresponding function key on the function keyboard. The system then performs the desired action, and makes available to the user all the options which are listed at the next level under the node selected. The user is also given the option of returning to any legal higher order node. Figure 3 gives a diagram of the systems organization.

The WPS Filing System

The user generally starts his analysis with a file of data containing many digital waveforms. In the course of analysis, (editing, transforming, etc.) of this data, he creates and modifies many new data files. To process all this data systematically requires the WPS to have a data filing system which can create, modify, delete, and retrieve mass storage data files. The WPS Filing System is the software which handles all accesses to the mass storage device.

It has complete responsibility for data handling which includes the formation of the file tables, and the associated bookkeeping functions.

The filing system allows dynamic assignment of names to any definable data set, which then can be stored and recalled using only the assigned name. The user can partition or subdivide one data file into two or more files or, if he wishes, union or merge two or more data files into one file. The filing system also allows the user to build new files by the arbitrary selection of data from existing data files. In addition, the user can delete newly created files if the results of a particular transformation are not promising.

A provision is available which will enable the user to choose a subset of the waveforms to be used in computing a preliminary set of transforms. If the results indicate that the transformation is useful, the system will return and process all of the waveforms; if not, the partial file will be deleted.

The filing system can record the sequence of promising user selected applications programs with the appropriate parameters so that the WPS can recreate any such sequence automatically on a new data set.

The filing system is also able to handle vector data files which are created as a result of a feature extraction process. All the features extracted from the source data set directly or through a series of transformations are placed in the same vector data file.

The WPS Graphics Software

The graphics software interfaces the user to the WPS via the on-line interactive display console. The user can analyze graphic representations of his source data and transformations of it, and direct the WPS to perform specified operations on his data via light gun and keyboard actions.

The graphics software provides the user with the capability to choose the most efficient presentation for a particular set of data. The display options included in this module augment the specific fixed format displays which present the results of the individual operations which are performed in the edit, transformation and feature definition modules. Approximately twenty options are provided, including both single and multiple waveform display formats. A complete listing of these options is given in Figure 3 under frames 09-00 and 09-01.

The Applications Programs

The Applications Programs are routines or algorithms which perform mathematical and statistical operations on the "current data set." These programs are not resident in core, but are stored in the Applications Program Library on a random

access storage device. Each program in the library is divided into segments or overlays, the number of which is determined by the size of the program. Small programs can be stored in one segment. After an applications program has been selected by the user, the system will search the library directory for the program's location on the storage device. When located, the first segment of the program is loaded into core and control is transferred to its entry point. The remaining overlays will be loaded upon request by the overlay currently in core. After completion of the selected program, control is transferred back to the WPS Executive along with a pointer to the output data file.

The applications programs provided to the user by WPS can be functionally grouped into three main modules: editing procedures, transformations, and a feature definition language. Each of these modules will be summarized below.

The editing procedures provide the user with the ability to edit digitized waveforms in order to accomplish event detection, artifact removal or segmentation of waveforms. Editing becomes very important in the case of long duration signals, but may also be relevant when processing short duration waveforms.

To accomplish these functions, the analyst is provided with algorithms for time alignment, deletion of intervals, and replacement of intervals. He will have the ability to create his new data base by manual indication (via the graphics terminal) of the beginning and end segments of pertinent regions of waveform data, or by on-line thresholding using the following criteria (partial list) where parametric values can be specified by the user: amplitude levels, average value within a time window, and cross correlation or convolution with a prototype or reference digitized waveform. A complete listing of these options is given in Figure 3 under the Edit Frame 09-00-12 and the Segmentation Frame 09-03.

The set of transformations can be subdivided in many ways. One subdivision which is pertinent when considering the data management aspects of the WPS is to subdivide each of the various waveform transformation algorithms according to the form of the data resulting from the application of the transformation. This method of subdivision results in two classes: (1) waveform to waveform operations, and (2) waveform to vector operations (e.g., waveforms to digital features where a single scalar is a special case).

The following transformations are included:

Basis Function Expansions

Spectral Analysis

Calculus-Algebraic Type Operations

Digital Filtering

Basis function expansions can be used to map the waveforms being analyzed into a new domain where the discriminatory information may be more apparent, or a subset of the calculated coefficients could be used as features for discrimination. The eigenvectors and discrimination vectors transformation[9] (options 07 and 08 of the Waveform to Waveform Transformation Frame 09-02 of Figure 3) are data dependent. All the expansions are "global" in the sense that any one coefficient depends upon the entire waveform. In problems where local information is significant, these transformations may only serve to make discrimination more difficult. Under the Algebraic/Calculus Frame 09-02-02 of Figure 3, the analyst will have the ability to form sequences of the operations listed, thereby giving him an extremely large transformational capability. For example, although the integral of the absolute value of the waveform is not explicitly listed, the analyst will have the ability to calculate it by combining the operations of rectification and integration.

The system includes a language, called the On-Line Waveform Processing Language (OLWPL), which can be used by the analyst to construct his own algorithms for waveform processing and feature extraction.

A desirable property of the language is that it permits the user to both define what he observes to be a good feature, and then test his hypothesis in a timely interactive manner. Hence, OLWPL has been designed to be a high-order language (a cross between FORTRAN and BASIC), thus eliminating lengthy and laborious programming on the part of the on-line user. On the other hand, it has enough low-level capability to allow the user to describe his hypothesis without the cumbersome manipulation of very high level operators. Thus, OLWPL will contain statements for normal arithmetic and logic operations, and facilities for handling waveforms and complete data trees without detailed input/output specifications from the user. Hence, it will be only necessary to identify a tree by name, or a waveform by its tree name, node name and identification number. The user will not have to supply parameters indicating the length of a waveform, how many waveforms are in a data tree, etc.

On the high level, many useful waveform processing operations will be available as subroutines that can be used as high level instructions. Initially, 36 built-in callable subroutines will be implemented. Provisions are included to allow the user to construct his own subroutine, name it, and enter it into the system such that it is then callable by name also.

4. The On-Line Pattern Analysis and Recognition System (OLPARS)

OLPARS is an interactive, graphics oriented, computer system for the solution of pattern analysis and pattern classification problems. The OLPARS system can be characterized as follows:

(1) It is a software system which allows a human operator to analyze digital preprocessed data (vector data) to determine the structure of the data and design pattern classification logic.

(2) It is implemented on a general purpose computer coupled to an interactive graphics display console.

(3) It requires that the input data consists of 100 or fewer digital measurements per sample.

It should be stressed that OLPARS is not a pattern classification system; rather it is a research tool which is used to design and evaluate pattern classification systems. The general purpose computer contains a library of pattern analysis and pattern classification procedures. By means of the graphics display console, a human operator can analyze his data, and based on what he sees, coupled with any "a priori" knowledge he may possess, choose an appropriate pattern classification procedure, observe the results and continue to iterate in this manner. Eventually one of two things will happen: (1) he solves the particular pattern classification problem he is working on, whereby the output of the computer consists of the design parameters for an automatic classifier which can then be implemented in the form of special purpose hardware or software, or (2) he cannot solve the problem. In this case, he has determined that his input data was inadequate to discriminate between the classes he wished to automatically identify, and he must return to the feature extraction or data collection phase.

As previously mentioned, OLPARS was initially implemented at RADC on a CDC 1604 computer coupled to a Bunker Ramo BR-85 display console. This vintage - 1957 computer equipment is no longer in operation at RADC. OLPARS is currently resident on two computer graphics systems at RADC. One version is on the PDP-11/45 computer under WPS, which uses the Vector General graphics terminal. The second version of OLPARS is implemented on the HIS 6180 computer under the MULTICS operating system. MULTICS is a time-sharing system that utilizes a virtual memory concept. Interactive graphics capability is provided by a Tektronix 4002A storage tube with alphanumeric keyboard, joystick and hardcopy unit. Since both systems are fundamentally the same with respect to the application software provided, we will first present a general functional overview of OLPARS which is implementation independent. Once this has been discussed, we will highlight the main differences between the PDP-11/45 OLPARS and MULTICS/OLPARS.

Functional Overview

OLPARS permits the system user to dynamically restructure the vector data files. The vector data structure is represented within OLPARS as a hierarchical tree where each node corresponds to a list of vectors. Partitionment of a list of

vectors is represented by branches to lower order nodes emanating from the node corresponding to the original list, with each subnode being associated with a sublist. The OLPARS user can select for processing the data associated with any node(s) by designating that node(s). Throughout the entire system, the concept of a "current data set" is used. Thus, the system will continue to operate on the latest data that the on-line user has designated unless specifically told to do otherwise. The OLPARS filing structure will allow continued arbitrary partitioning.

In addition to the above operations, new data trees may be created when the current data set is operated on by a linear transformation, a different partitionment of the data is desired, or a new data tree may be created by performing logical operations on selected nodes of a specific tree. The operations of union, intersection, complement of union, and complement of an intersection can be applied to the selected data sets. When a transformation is applied at the topmost node of a tree, the structure below the node is maintained, and the transformation is applied to all the data vectors. A transformation may be selectively applied to the data below a specified node in which case a new tree is generated, involving only the data corresponding to the selected node.

We can functionally group the current OLPARS options into the following categories: system utility options, data management, data display, structure analysis, feature evaluation, data tree transformation and classification logic design and evaluation. Included among the system utility options are routines to print pertinent data characteristics (such as the selected data set vectors or the selected data set tree structure) and statistics (including data class ranges, measurement overlap between classes, covariance matrix for each class, etc.). The user can also create a random test data set from the current data tree, display a logic tree or the current data tree, and list the data trees in current active storage.

Data Management

The data management routines include options for data input/output, data tree modification, data storage and data printout. The options for data input/output and data storage will be discussed later, since many of them are implementation dependent. The data tree modification options automatically restructure the data into the modes defined by the on-line user. These include the ability to add a data class to the current data from other existing data trees, modify a tree name or data class name, combine data classes, create a data tree from existing data classes, and delete a data class from a data tree. In addition, options exist to remove a data tree from storage, delete a subnode structure, and remove data vectors from a data tree. Finally, a user can create subnode structure via partitionment of a data projection display or use of boolean (linguistic) statements.

Data Display

OLPARS provides the user with the capability to project a data set into a one or two space representation. Extensive facilities for manipulation and modification of these data projection displays are available. These include the ability to modify the bin size of a histogram, draw or remove a partition on a data projection, change the data class composition on a two space projection, identify selected data points, change scale, and draw a logic design boundary. There exist several other options available to the user when the current data set contains more vectors than can be displayed on the display screen for two space mappings.

Structure Analysis

As previously mentioned, the pattern analysis problem arises as a prerequisite to solving pattern classification problems. The solution to the pattern analysis or structure analysis problem consists in the determination of the natural or inherent distribution of vector data in feature space via the identification of clusters, i.e., groups of vector data samples which are closely related by some metric. The basic use of structure analysis in OLPARS is to determine whether the data for a particular class is unimodal or multimodal. If it is determined to be multimodal, one can then subdivide the class according to its modes before proceeding to design classification logic. One of the truly powerful capabilities of interactive systems such as OLPARS is the capability to take advantage of the human ability to visually investigate data structures, and interactively partition vector data sets.

All of the algorithms for structure analysis in OLPARS rely upon the human projecting the data onto a one or two spaces and visually observing the structure. He can then partition the data into subclasses (create subnode structure in the data tree) via use of boolean (linguistic) statements or piecewise linear boundaries drawn on the data projection display.

The user may perform a projection of data into a one or two space defined by the following projection axes: arbitrary vectors, coordinate vectors, eigenvectors or Fisher discriminant vectors. Arbitrary vectors are those chosen by the user. They may be manually input or retrieved from system files. Hence, they may be calculated within OLPARS or external to the system. The coordinate vectors are the axes defined by the features obtained from the feature extractor. The eigenvectors used for data projection in OLPARS are computed from the lumped data covariance matrix. The user chooses the eigenvector(s) he wants by choosing the corresponding eigenvalue(s).

By the Fisher discriminant vectors are meant the Fisher Linear Discriminant d_1, and a second vector d_2, where d_2 is that direction which maximizes the projected between-class scatter relative to the sum of the projected within-class scatter

under the constraint that d_2 be orthogonal to d_1.[10] If the one space option is chosen the data is projected onto d_1. Options exist for choosing the two classes upon which the projection is based. The two classes may consist of any two classes of the current data set, or they may be composed of any two arbitrary groups of classes which are lumped together, where each group is considered as one class for the purpose of the above calculation. These groupings need not comprise the entire data set. However, the entire data set is projected on the resulting Fisher discriminant(s).

In MULTICS/OLPARS an additional data projection display is available, which is called the Nonlinear Mapping (NLM) Algorithm.[11] The NLM algorithm is based upon a point mapping of N L-dimensional vectors from L-space to a two-dimensional space such that the inherent structure of the data is approximately preserved under the mapping. The approximate structure preservation is maintained by fitting N points in the two-dimensional space such that their interpoint distances approximate the corresponding interpoint distances in the L-space.

Feature Evaluation

In solving a pattern classification problem, the researcher will often be concerned with the discriminatory qualities of the extracted features. In general, it is desirable to use the minimum number of features to achieve a satisfactory solution. To this end, OLPARS provides two methods for ranking the discriminatory power of a set of L features. An optimal method for ranking the L features must consider the decision logic criterion, such as the Bayes Risk or the probability of error. This, in turn, requires the estimation of the joint probability functions for all possible n-tuples. The obvious computational difficulties in obtaining an optimal ranking preclude this approach in all but the simplest problems. Therefore, two sub-optimal algorithms are provided as options to rank order the L features $x_1, x_2, \ldots, x_L$. Each algorithm provides three distinct types of rankings. The first uses a significance measure of a particular component, say x_p, for discriminating class i from class j. The second type of ranking uses a significance measure of x_p for discriminating class i from all other classes. The last type of ranking uses a measure of the overall significance of x_p for discriminating all classes.

The first measure is called the Discriminant Measure. It is particularly useful for ranking the L features when the class conditional probability distributions are approximately unimodal. It essentially measures the ratio of the squared difference between the estimated class means to the sum of the estimated class variances along the feature being evaluated for a user specified pair of classes.

The second measure is the Probability of Confusion Measure which is based on a histogram estimation of class conditional probabilities. The values produced are measures of the overlap of these probabilities. Hence, the smaller the

value, the better the measurement. User interaction is designed to allow selection of the interval range and number of histogram bins which will represent the data distribution. Computationally, it is much more complex than the previous measure. It is recommended for use when the unimodal assumption cannot be justified.

Data Tree Transformation

There are three options available in OLPARS for data tree transformation. Upon execution of any of the transformations, the system applies the transformation to every data vector in the current data set and creates a new data tree within the filing system. However, the structure of the old data tree is preserved under the transformation so that the new data tree looks exactly like the old one, the difference being that the data represented by the new tree has been transformed.

The three data transformations provided are eigenvector projections, a normalization transformation, and measurement reduction. When the eigenvector option is selected, the system computes the eigenvectors of the estimated lumped covariance matrix. The user then has the option to project the current data onto an M-dimensional eigenvector subspace by selecting the M eigenvectors corresponding to the M largest eigenvalues. The resulting M-dimensional subspace provides a least squares fit to the current data set. The normalization transformation creates a new tree whose features correspond to those of the current data set divided by the standard deviation of that feature. Hence, each feature of the new data tree will have unit variance. By means of the measurement reduction option, the user can project the current data set onto a coordinate subspace. His choice of subspace is based on the results of the two feature evaluation procedures discussed previously. Based on the feature rankings of either of these algorithms, the user can select a subset of the original features to define a coordinate subspace, and hence, the desired linear transformation.

A fourth method for data transformation is available in MULTICS/OLPARS. This additional option is a feature compiler which makes use of the MULTICS PL/1 compiler. This feature compiler allows the analyst to define a new data tree whose features are arbitrary arithmetic combinations of the features of the current data set. The user accomplishes this by constructing a PL/1 program on-line which defines the features of the new data tree in terms of the features of the current data set. The OLPARS routine then calls the MULTICS PL/1 compiler to compile the user defined transformation, and then executes this code to create the new data tree.

Logic Design and Evaluation

The OLPARS logic design facilities provide extensive mathematical/graphical

procedures for allowing the user to tailor classification logic design to the structure of the class data. As previously mentioned, the general philosophy of OLPARS is that pattern classification operations are preceded by structure analysis to insure that each class is unimodal. Although not always required, the unimodal property is highly desirable in order to insure an effective logic design. When multimodal class data has been subdivided into unimodal subclasses using structure analysis options, OLPARS provides the capability to reidentify the decision regions for each of these subclasses with the original multimodal class label upon completion of the classification logic design.

Upon selection of a logic design option, a logic tree is initialized by the system with a single node consisting of all the lowest order data classes of the current data set. The system keeps a record of the decision logic as it is created. The actual form of the logic constructed is that of a hierarchical tree where each node corresponds to a partial decision. The logic design facilities provide the capability to create/display a logic tree, modify a logic design and evaluate a logic design.

OLPARS provides three basic techniques for designing classification logic: nearest mean vector logic, Fisher pairwise discriminant logic, and between group logic. Nearest mean vector logic is a K class classification technique which classifies an unknown vector in the feature space according to a metric computed from the unknown vector to the mean vectors of the K classes of a design set. The decision is for the class which produces the minimum value of the metric. In OLPARS the user has the choice of three metrics plus the capability of specifying a reject strategy under each. The three metrics provided are the Euclidean distance, weighted vector distance, and the Mahalanobis distance. For the weighted vector distance, the Euclidean distance along each feature is weighted by the inverse of the variance along that feature. For the Mahalanobis distance, the Euclidean distance is weighted by the inverse of the covariance matrix. The optional reject strategy allows the user to reject an unknown vector if its distance from each class mean is greater than some specified value. A separate reject distance may be specified for each class.

Fisher pairwise discriminant logic is constructed by computing the Fisher linear discriminant with appropriate thresholds to distinguish between every pair of classes (subclasses) within a designated group. Once the within group pairwise classification is complete, the pairwise decisions are combined to produce a final decision. The group of classes (subclasses) might be the original K classes (subclasses) of the current data set, or the group might be composed of a subset of K. In the case where the user does not subdivide the K classes (subclasses) he would compute $K(K - 1)/2$ pairwise discriminants. The output from each pairwise discriminator consists of a vote for one of the two classes being discriminated (or a vote to reject the unknown vector if the user desires to establish a reject region). The vote count for each class (and the reject region, if it exists) is collected, and the final decision is for the class

(including the reject class) which received the maximum vote count, provided this maximum is greater than or equal to a user specified value. If the maximum vote count is less than this specified value, the unknown vector is rejected. As implied above, the user can select any one of four different threshold options to be used in each pairwise discriminator. These allow the existence of various reject strategies or none at all.

Once a Fisher pairwise discriminant logic has been constructed, OLPARS provides the user with the capability of individually modifying each of the class pair logics. The possible changes that can be made to each logic "box" are to modify the Fisher logic, or to replace the existing logic. Allowable modifications of the Fisher logic include changing the number of thresholds (change threshold option), moving the threshold(s), eliminating features from the calculation of a specified discriminant, or inserting a user defined boundary in the Fisher discriminant plane. The existing logic of each box can be replaced by an arbitrary one-space discriminator, by drawing a boundary in an arbitrary two-space discriminant plane, or by means of a Boolean (linguistic) partition.

An obvious drawback to computing all K(K-1)/2 pairwise discriminants is the potentially large number of combinations. In most problems of interest some of the classes are statistically disjoint and quite easily separated from one another. If these disjoint class groups can be identified and logic designed to discriminate the groups, then the pairwise discrimination need only be computed for the statistically overlapped classes within the group. Since the OLPARS user will not generally know "a priori" how the classes are distributed in feature space, an option is provided (between group logic design) to allow the user to detect nonoverlapping groups of classes, and draw a separating piecewise linear boundary on the display to partition the feature space.

Under between group logic design, the analyst actually participates in the logic design process. He has the capability to interactively construct his own classification logic tree. He is not constrained to choose a preprogrammed classification procedure, or to follow any predetermined logic structure. At any given node in the logic tree, the user can partition the data present at that node by defining his own boundaries in an arbitrary one or two space projection, or by means of a Boolean defined partition. However, at any subnode of the logic tree, the user may also call upon the nearest mean vector or Fisher pairwise logic, which were previously discussed, to perform a complete within group classification for that subnode.

All of the one and two space projection options available for structure analysis are also available to the user for group logic design. Hence, the user can project class data onto the Fisher discriminant plane(s), eigenvector plane(s), coordinate plane(s), and arbitrary plane(s). For one space logic, the vector to be classified is projected onto a user specified vector direction, and the value of this scalar (dot product) is compared to the value of the user defined

threshold (boundary). For two space logic, the user has the capability of defining the two space onto which the data is to be projected, and then drawing up to two piecewise linear convex boundaries having up to five linear segments each as a means of defining the decision boundary. In addition, OLPARS provides for the implementation of a user defined linguistic logic partition. In MULTICS/OLPARS, the user can write any Boolean statement (one that can be evaluated as true or false) provided it is a legal PL/1 statement, and then use this statement to define a partition.

Under the classification logic design and evaluation facilities, temporary logic evaluation results are displayed following any logic implementation. Upon completing the logic design, the user can next evaluate the design against any data set (test set) and review the results of that evaluation by means of a confusion matrix format. Adequate logic may be output to the system printer or stored within OLPARS. Logic which does not provide adequate discrimination may be supplemented, modified or deleted. This completes the functional overview of OLPARS.

Comparison of two Implementations

We will now briefly contrast the two implementations of OLPARS which exist at RADC. The version on the PDP-11/45 computer is a subsystem under WPS. It is a single user (dedicated) system employing high performance CRT interactive graphics (Vector General graphics terminal with three dimensional rotation, translation and scaling of the display image, light pen, data tablet, alphanumeric keyboard, function keys and intensity modulation). As a module under WPS, PDP-11/45 OLPARS provides for ease of interaction between the feature extraction mode conducted under WPS, and rapid testing of these hypotheses under OLPARS. However, since this system is built on a mini-computer, there are core limitations in terms of the size of the data base which can be processed. It is witten in assembly language. The options available to the OLPARS user are set up in a hierarchical tree control structure (see Figure 4). At any point in the system operation, the current options available to a user are represented by a menu which is displayed on the lefthand side of the CRT odisplay. The user can select an option by depressing the corresponding function key on the function keyboard. The system then performs the required action and makes available all the options which are listed at the next level under the node selected. The user is also given the option of returning to any legal higher node.

Since the PDP-11/45 OLPARS is a module under WPS, data storage is provided by the WPS filing system. The WPS filing system has facilities for handling both waveform and vector data files. OLPARS can store and retrieve data from the vector data files only. Vector data for OLPARS processing can be input into the filing system from magnetic tape, or created by feature extraction algorithms in WPS. In the latter case, waveform to vector data transformations in WPS create a vector data file in the WPS filing system, thus providing a direct

communication link between the two systems. Data and programs are overlaid and stored on a ten million word disc. Data swapping is handled in software as opposed to hardware as is the case in MULTICS/OLPARS. There is no limit to the number of trees which can be stored, other than the physical limitation of the size of the disc.

The WPS system software provides a background/foreground processing capability. Hence, a PDP-11/45 OLPARS user can execute a time consuming non-interactive job in background and continue to interactively work in the foreground mode. Data and logic trees can be output on magnetic tape. New options can be readily added to the system; however, they must be written in assembly language, and a program overlay built and added to the system by one knowledgeable of the WPS system software.

MULTICS/OLPARS has a distinct advantage over the PDP-11/45 OLPARS in terms of storage capacity (virtual memory), ease of data access, multi-user environment, and data base sharing among users. Besides providing more advanced pattern classifier logic design capability, the system will be available to other government agencies and their defense industry contractors by remote access through the ARPA computer network. It is written in PL/1. Interactive graphics is provided by means of a storage tube (Tektronix 4002A with alphanumeric keyboard, joystick and hardcopy unit). There is no control tree structure for user options. The MULTICS/OLPARS user is free to select any option at any time by typing a 4 to 8 character option label. Through MULTICS the user can make use of an absentee (batch) job capability. Thus, a sequence of OLPARS options which are lengthy computationally and require no interaction can be submitted for execution at a later time.

For data storage MULTICS/OLPARS makes use of the existing file facilities contained in MULTICS. Each user is provided with a temporary data storage area as well as a set of more permanent data files. The temporary area contains his current system description and his current data tree. His permanently assigned area provides file entries for data which may be utilized on a day-to-day basis as well as a hardcopy dump area for delayed printout. In addition to the permanent user area, the central system contains the object programs available under MULTICS/OLPARS and a data storage area from which data may be transferred into any user's temporary data area. Under the MULTICS structure, each user has access to the programs in the central system directory for operations upon data in his own temporary storage area. Source programs for MULTICS/OLPARS are also stored in the central system directory. System programmers may add to and/or modify programs in MULTICS/OLPARS in PL/1 by means of MULTICS system functions to produce new or revised object versions within that directory.

Data may be brought into current storage and formatted for MULTICS/OLPARS usage in a variety of ways. Currently, procedures have been implemented which will accept data from cards, magnetic tapes and other MULTICS files. Permanent

storage files may be maintained either for the exclusive access of a particular user or for common access by a number of analysts. Data trees may be outputted to either type of storage area, retrieved and deleted. In addition, classification logic and projection vectors may be stored, retrieved and deleted from exclusive user storage. Current data storage facilities provide for immediate access to any of up to 20 data trees. Once in current storage, a data tree can be modified by any of the data modification options previously described. Data trees from current data storage can be permanently stored on magnetic tape.

The major differences between the two systems with respect to algorithms for structure analysis and pattern classification have resulted because of storage limitations on the PDP-11/45 system and the power of the MULTICS operating system. Options only available on MULTICS/OLPARS include the nonlinear mapping algorithm for structure analysis, the use of Boolean (linguistic) logic statements for partitioning data trees in structure analysis and as a feature compiler for data transformations, and the ability to eliminate measurements for selected Fisher pairwise logic "boxes." In addition, MULTICS/OLPARS allows the creation of independent reject strategies. Any final classification node of the logic tree may be appended with a Boolean reject strategy. A vector classified at a node and evaluated as false by the strategy will be rejected.

5. The Other Elements of the Laboratory

The major elements of the RADC Interactive Laboratory for the Design of Pattern Recognition Systems are WPS and OLPARS which were previously described. In addition, it contains an analog data processing capability, a feature extraction software system, and a long waveform analysis system. Each of these remaining elements will be briefly described in this section.

The Laboratory has an Analog Data Processing configuration to complement its digital processing capability resident in the PDP-11/45 computer system. The nucleus of the analog configuration is an Applied Dynamics A/D-5 analog computer. This unit provides a 100 amplifier system, together with function generators, logic, analog to digital converters, digital to analog converters and numerous other options all under digital control. The A/D-5 has been interfaced to the PDP-11/45 digital computer to provide a hybrid processing capability. To further enhance the system, analog tape units, a spectrum analyzer, correlation and probability analyzer, switchable filters and various other analog instrumentation units have been integrated to make this a complete, cohesive and extremely powerful, yet versatile system. The combined A/D-5 - PDP-11/45 system provides the capability to begin with raw analog data, particularly for pattern recognition problems, pre-process it in analog form, convert it to digital data, process it digitally and present it to the user via a high performance interactive graphics system.

The Hybrid Feature Extraction Software System (FESS) is implemented on a hybrid system consisting of the PDP-11/45 central processor, the A/D-5 analog computer, the Tektronix 4002A display and other peripherals.[12] The main purpose of FESS is to generate a large data base of features from analog data after the features have been defined on WPS. This large data base can then be used in designing the classifier on OLPARS. Part of this data is used as an independent test set for testing the designed classifier.

Fifteen feature extraction algorithms are currently included in the system. The use of these algorithms is interactive in the sense that parameters must be specified by typing them in at the Tektronix keyboard at the request of the system. The parameters are known by the user as a result of the feature definitions as defined by use of WPS. The actual extraction of the features by FESS is accomplished by analog processing. The menu of features is at present limited to those which have been chosen by experience on previous problems. Some examples of these operations include: spectrum analysis, filtering, Laguerre and Legendre expansions, peak locations and zero crossings, auto and cross correlations, and nonlinear functions approximated out of piecewise linear functions of the waveform which can be constructed by a diode array.

The Long Waveform Analysis system is an interactive software system designed to digitize and display analog data.[13] It is implemented on a PDP-11/45 computer with an analog to digital converter, tape units, a time code reader, a disk and a Tektronix 4002A display with hard copy.

The main purpose of the Long Waveform Analysis system is to be able to observe very long waveforms, and perform spectral analysis upon them. Data from up to 99 lines of a time domain waveform with up to a 2048 data point window per line can be displayed on the storage tube without the objectionable flicker rates of the Vector General display. Typically only up to 20 lines of data are used. In spectral analysis, the proper Nyquist sampling rate can be interactively determined.

This expandable system currently consists of two interactive programs. The first program requests the user to type in a number of parameters which are used to search one of the analog tape units for a designated starting time code. After finding the data with designated starting time, the system digitizes the data at a rate determined by the user and stores this data on a disk. The data can be analog filtered prior to digitization by one of several filter transfer functions. The second program contains display options and has access to the data which has been stored on the disk. The data can be displayed either as a time waveform or as a power spectrum on the Tektronix 4002A. Various scaling and blanking options enable the user to examine details of power spectrum and time domain waveforms.

6. Applications

Elements of the current laboratory have been used on several data sets representing various problems to design classifiers. For the applications described below, the Waveform Processing System was not available so that features were determined and defined by observing a hard copy library of waveforms and their Fourier transforms obtained from a storage tube. The classification based upon these features was then interactively obtained using OLPARS. Table 2 shows empirical results obtained on a number of selected problems of this type.

ORG	SENSOR	OBJECTS	C	F	S	P(C)	REF
RADC	Geophone	Vehicles	5	44	1322	.85	14
RADC	Geophone	Vehicles	5	33	1322	.85	14
RADC	Geophone	Vehicles	5	16	1322	.85	14
PAR*	Microphone	Vehicles	4	36	1328	1.00	15
RADC	Photometer	Space Objects	3	13	252	.96	
RADC	Electro-cardiac Probe		2	10	2222	.97	
PAR*	Image Scanner	Hand Print Characters	15	45	100,000	.99	16
NASA	Multi-spectral scanner	Crop Types	7	12	847	.97	
RPI**	Medical Analysis Application						

* Pattern Analysis and Recognition Corp., Rome NY
** Rensselaer Polytechnic Institute, Troy NY

Table 2 - Selected Applications of the RADC Laboratory

A legend of the abbreviations used in Table 2 follows: ORG is the organization who obtained the results, C is the number of classes, F is the number of features, S is the total number of data samples, P(C) is the estimated probability of correct classification, and REF is the reference publication for the given results.

In addition to designing classifiers, OLPARS has been used to test the usefulness of a proposed set of features generated external to the laboratory. This is done by designing in software a classifier on OLPARS using the proposed features and observing its performance. If the performance is low, it is assumed that new features are needed. In other applications, elements of the laboratory have been used for data analysis where classification is not the final objective. Examples of this type of application include analysis of medical data dealing with shock trauma to construct procedures for screening patients who would most profitably benefit from treatment under conditions of limited medical personnel.

It has been proposed that features useful for speech classification could be transmitted in speech communication problems, to obtain bandwidth compression in vocoders. Only preliminary results on this application are available thus far.

A copy of an earlier CDC 1604 version of OLPARS[17] exists in the Department of the Navy and has been used by them and some of their contractors.

7. Educational and Training Aspects

Widespread usage of the RADC Interactive Laboratory for the design of Pattern Recognition Systems is advocated and encouraged. To date, numerous individuals and organizations which include universities, industries and Government laboratories (Air Force, NASA, Army, etc.), have successfully used the system to aid in the solution of their diversified problems ranging from medical diagnosis to crop classification. In such cases, the individuals usually obtain copies of the relevant reports describing the system and its software first. They then arrive at the Laboratory a day earlier to become acquainted with the system prior to actual operation on their problem. In most cases, this has worked satisfactorily with the time spent averaging about three days. Usage of the equipment by other Divisions within RADC continues on a regular basis. Support and assistance is provided by personnel of the Information Sciences Division of RADC.

For more general exposure to the field of Pattern Recognition and the relationship of the Laboratory to this field, short 1/2 day seminars were offered in earlier years. More recently, a formal in-house course was offered by one of the authors (Prof Gerhardt) during the Fall of 1973. The first portion of the course, attended by RADC personnel, stressed the different approaches to Feature Extraction and Pattern Classification. The text, "Introduction to Statistical Pattern Recognition", by K. Fukanaga was used. Assigned problems and individual projects primarily involved the use of OLPARS. In this way, the participant gained a working knowledge of not only the basic tools and the hardware and software, but of the application of the system to areas related to his specific field of interest. Data sets from the text were used and imbedded in a variety of different problems. As examples, some of the results obtained by each participant included the plotting of the data in coordinate, principal eigenvector, and Fisher Discriminant space, linear classifier design, and piecewise linear classifier design among others. Applications included radar classification, speech recognition and communications.

More recently, in April 1975, two, two-day workshops directed to industry and other Government agencies were offered by RADC personnel. These provided a broad overview, and discussions of usage and applications. It is intended to follow this with a course similar to the one mentioned above to provide others outside RADC with a similar working knowledge of the Laboratory system.

Hundreds of groups and individuals have visited RADC's Interactive Laboratory. These have included visitors from as far away as Europe and Japan, as well as graduate students from local universities interested in the field of Pattern Recognition and Signal and Image Processing. It is hoped that these workshops and courses involving the laboratory will continue to encourage more widespread use of the Laboratory. Anyone interested may contact the authors directly for more detailed information.

8. Sample Size in the Empirical Approach

One point that is frequently overlooked when taking an empirical approach to classifier design is insuring an adequate data base of class representative samples. It is clear that if class conditional densities exist for all classes, the probability of exact equality of any two samples is zero, if computer roundoff error is neglected. Hence, under the above assumption, given a finite set of samples, any subset can be separated from any other subset. There is nothing but patience, ingenuity, and complexity of the classifier that limits one's ability to do this. Thus, one can construct a statistical trap if he is not careful, by thinking he has obtained better results than he has. If indeed the design is "tuned up" for one set of samples of the population, it is likely to do worse on another finite test set of samples.

Foley [18] has shown that in a two class classification problem under the hypotheses of Gaussian class conditional densities of equal known covariance matrices, the use of estimated sample means and Fisher's linear discriminant as the classifier, that a good rule of thumb is that the ratio of the number of vector samples to the number of features in the design set should exceed 3.5 per class. If the number of data samples used for testing the classifier is equal to the number of data samples used in classifier design, the total number of data samples M needed under Foley's hypotheses is $M > 7LN$ where L is the number of features and N is the number of classes. It is surprising to note results in the literature where the amount of data does not satisfy either criterion. There is not yet a general definitive answer to this problem when Foley's assumptions are weakened. Some results under some weaker hypotheses have been obtained by Mehrotra. [19]

Acknowledgement

It would have been impossible for the authors to have written this paper without the help of the data provided by many individuals of the Information Sciences Division of the Rome Air Development Center and many of its sponsored contractors in this field, and a debt of gratitude is owed to them. Since the thrust of this conference is on interactive use of computers, graphics and pattern recognition, it was decided that the actual preparation of the written version of this paper would be done by interactive means. The authors acknowledge the help of Mr. Edward LaForge and Miss Roberta Carrier of RADC who

prepared the manuscript using the SRI developed oN-Line System (NLS) via the ARPA Network.

The work as performed by Professor Gerhardt was partially supported by the Air Force Office of Scientific Research (AFSC) under grant AF-AFOSR-73-2486 with the guidance of Lt Col Thomas J. Wachowski.

References

(1) L.N. Kanal, "Interactive Pattern Analysis and Classification Systems: A Survey and Commentary," Proc. of IEEE, Vol. 60, pp. 1200-1215, October 1972.

(2) R.L. Mattson & J.E. Dammann, "A Technique for Detecting and Coding Subclasses in Pattern Recognition Problems," IBM Journal, Vol. 9, pp. 294-302, July 1965.

(3) P.J. Martin, "User's Guide - DOCUS Pattern Recognition Overlay," Informatics Inc., Bethesda, MD., Tech. Rept. TR-67- 575-4, January 1967.

(4) J.W. Sammon, Jr., "On-Line Pattern Analysis and Recognition System (OLPARS)," Rome Air Development Center Tech. Rept. TR-68-263, August 1968.

(5) J.W. Sammon, Jr., D.B. Connell, & B.K. Opitz, "Programs for On-Line Pattern Analysis," Rome Air Development Center Tech. Rept. TR-71-177 (2 Vols.), September 1971.

(6) E.J. Simmons, Jr., "Interactive Pattern Recognition - A Designer's Tool," 1973 National Computer Conf. AFIPS Proc. Vol. 42, pp. 479-483.

(7) H.E. Webb, Jr., & D.H. Foley, "On the Design of Waveform Classification Systems by Interactive Man-Machine Methods," AGARD Conference Proc. No. 94 on Artificial Intelligence, pp. 29-1, 29-19, September 1971.

(8) D.H. Foley, H.W.Webb, Jr., A.H. Proctor, & J. C. Faust, "Waveform Processing System (WPS)," Rome Air Development Center Tech. Rept. TR-71-235, November 1971.

(9) D.H. Foley & J.W. Sammon, Jr., "An Optimal Set of Discriminant Vectors," IEEE Trans. Comput., Vol. C-24, pp. 281-289, March 1975.

(10) J.W. Sammon, Jr., "An Optimal Discriminant Plan," IEEE Trans. Comput. (Short Notes), Vol. C-19, pp. 826-829, September 1970.

(11) J.W. Sammon, Jr., "A Nonlinear Mapping for Data Structure Analysis," IEEE Trans. Comput., Vol. C-18, pp. 401-409, May 1969.

(12) P.J. Caruso, "Hybrid Feature Extraction Software," Rome Air Development Center Tech. Rept. TR-74-262, (3 Vols.), November 1974.

(13) A.H. Proctor & Capt D. White, "The Long Waveform Analysis System," Rome Air Development Center Tech. Rept., (To Appear).

(14) A.H. Proctor, J.E. Roach, & Capt M.H. Fick, "RADC Seismic Classifier Design," Rome Air Development Center Tech. Rept. TR-73-221, August 1973.

(15) J.W. Sammon, Jr., R.M. Oelsligle et al, "Time Domain Analysis for Inverse Scattering, Vol. II - Acoustic Signature Analysis," Rome Air Development Center Tech. Rept. TR-72-292, November 1972.

(16) J.W. Sammon, Jr., J.H. Sanders et al, "Handprinted Character Recognition Techniques," Rome Air Development Center Tech. Rept. TR-70-206, October 1970.

(17) J.W. Sammon, Jr., "Interactive Pattern Analysis and Classification," IEEE Trans. Comput., Vol. C-19, pp. 594-616, July 1970.

(18) D.H. Foley, "Considerations of Sample and Feature Size," IEEE Trans. Inform. Theory, Vol. IT-18, pp. 618-626, September 1972.

(19) K.G. Mehrotra, "Note on Probability of Error on Design Set," Rome Air Development Center Tech. Rept. TR-73-114, April 1973.

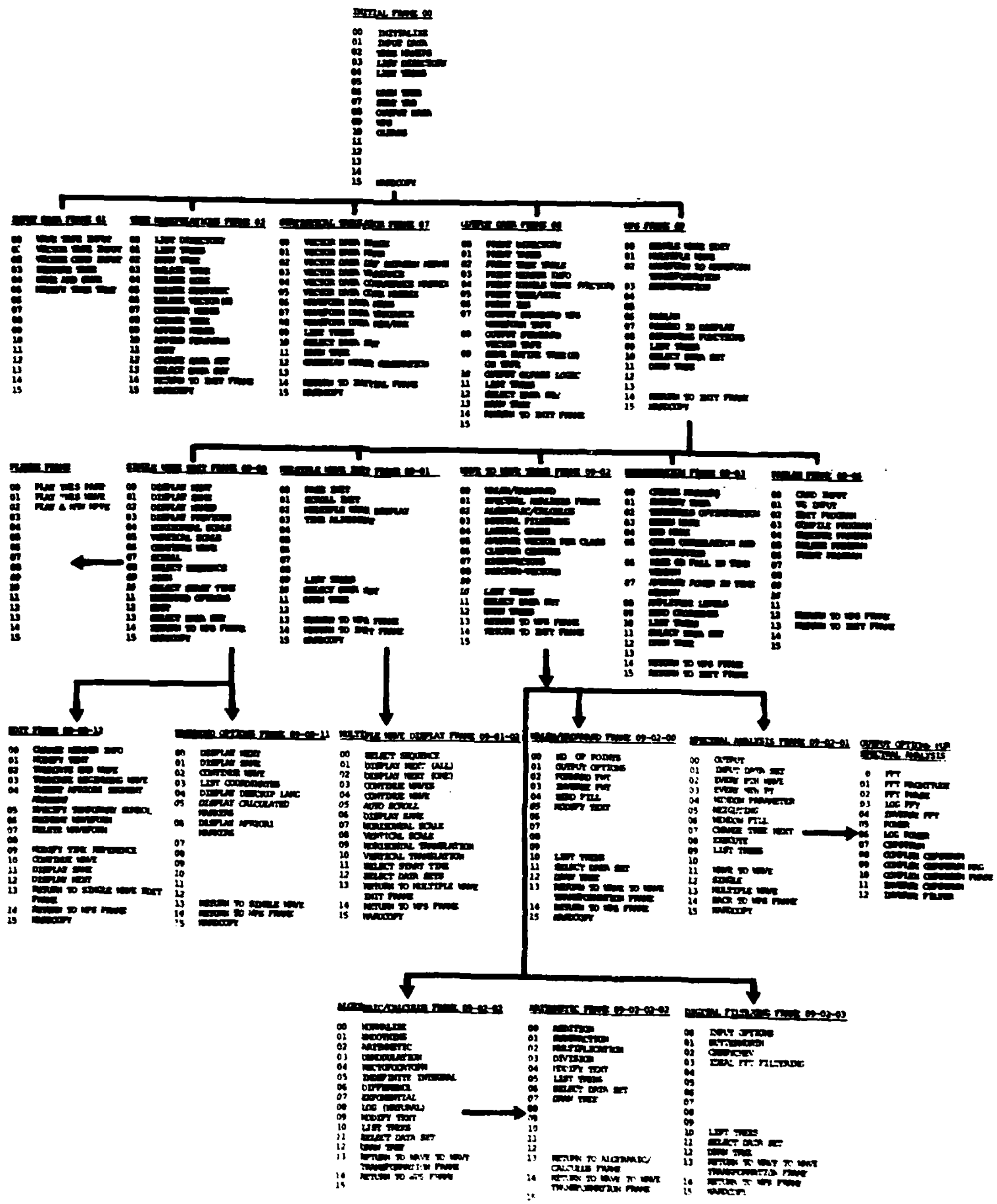

Figure 3 - WPS Control Tree

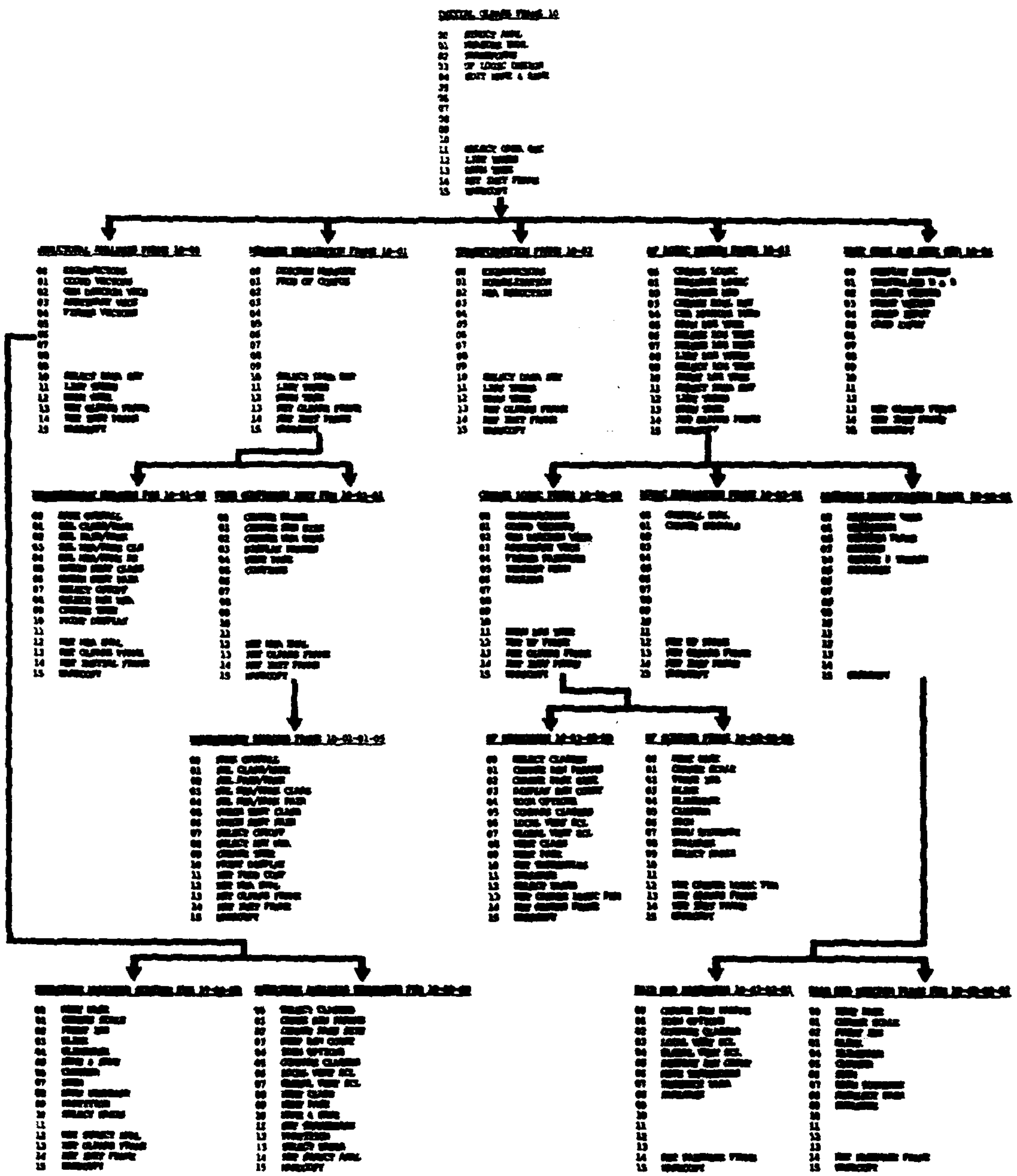

Figure 4 - PDP 11/45 OLPARS Control Tree